Using Evidence-based Arguments to Support Dependability Assurance – Experiences and Challenges

Janusz Górski

Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland; Tel: +48 58 347 1909; email: jango@pg.edu.pl

Abstract

The presentation introduces to the problem of evidence-based arguments and their applications. Then, based on the experiences collected during development and commercial deployment of a concrete solution to this problem (system NOR-STA) we overview selected challenges and the ways of addressing them.

Keywords: Evidence-based argument, assurance case, conformance case, tool support.

1 Introduction

The interest in using explicit evidence-based arguments with respect to socio-technical systems was growing over last forty years. It originated from the concept of *safety case* addressing the need to demonstrate safety and then was generalized to the concept of *assurance case* addressing a broader scope of objectives (like security, reliability, privacy). It has been also recognized that explicit evidence-based arguments can be used to demonstrate conformity with pre-defined sets of structured requirements of standards and other normative documents, which resulted in the concept of *conformance case.*

In this paper we briefly describe selected challenges which we were facing while developing, implementing and deploying the Trust-IT methodology and the NOR-STA system supporting applications of evidence-based arguments. NOR-STA has been gradually developed in a series R&D projects: EU sponsored projects DRIVE, PIPS and ANGEL, Polish-Norwegian Research Fund sponsored project ERM and European Regional Development Fund sponsored project NOR-STA. Since 2014 NOR-STA is a commercial product offered by Argevide, a spin-off company of Gdansk University of Technology [1]. More about challenges and the related solutions implemented in NOR-STA can be found in [2].

2 About evidence- based arguments

Argument is an attempt to persuade someone of something, by giving reasons and/or evidence for accepting a particular conclusion**.** This 'something' we want to argue about may be, for instance, assurance of some important property (like safety, security, privacy, reliability), conformance with a stated set of criteria (imposed by a standard, norm, directive, recommendation) or any other property selected as being of interest to the parties exchanging the arguments. An example argument could be:

> ***Module correctness argument:***
>
> *Tests confirm that this software module meets its requirements because test results are positive and the tests coverage is sufficient.*

Looking more closely to this argument we can identify two parts which are of different nature: the *logic part* and the *epistemic part*.

The logic part establishes the 'conveyance' relationship between the conclusion (also called *claim*) of the argument and the premises of the argument. In our example the claim postulates that '*the module meets requirements*' and the premise postulates the fact: '*test results are positive and the test coverage is sufficient'*. The 'conveyance' relationship between the two is established by the *strategy of argumentation* (the inference rule) which asserts that from the truth of the premise we can conclude the claim. It usually needs some *rationale* justifying the reasons for acceptance of such strategy. In our example the strategy of argumentation is: *argumentation by referring to test results and test coverage* and the rationale could be: *experience shows that positive results of tests of adequate coverage reliably demonstrate fulfilment of the requirements*. A graphical representation of the logic part of our example argument is given in Figure 1.

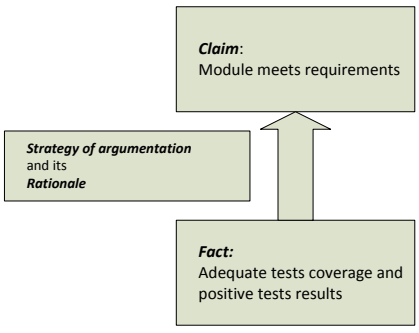

Figure 1 Logic part of *Module correctness argument*

The epistemic part of the argument focuses on providing *evidence* which in its broadest sense includes everything that can be used to determine or demonstrate the truth of the fact referred to in the argument. For instance, the fact *it is raining outside* could be demonstrated by a video stream from the camera looking outside through the window.

In our *Module correctness argument* example, such evidence could include the test plan and the test results for the considered software module. In our further considerations we assume that evidence is delivered in electronic form: text, graphics, image, video, audio, sensor measurements etc. A graphical model of the epistemic part the example argument is given in Figure 2.

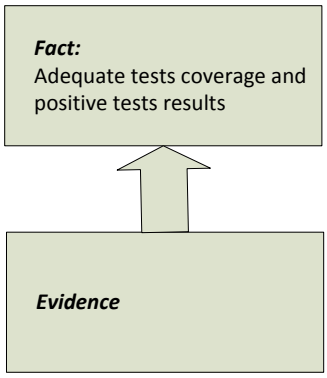

Figure 2 Epistemic part of *Module correctness argument*

In general, the premises of an argument can, in addition to facts, also include *assumptions* imposing constraints on the context of argumentation as well as more specific claims (*sub-claims*) which need further argumentation. This latter possibility results in hierarchical argumentation structures of an arbitrary depth. Possible extension of the *Module correctness argument* by introducing additional premises is illustrated in Figure 3.

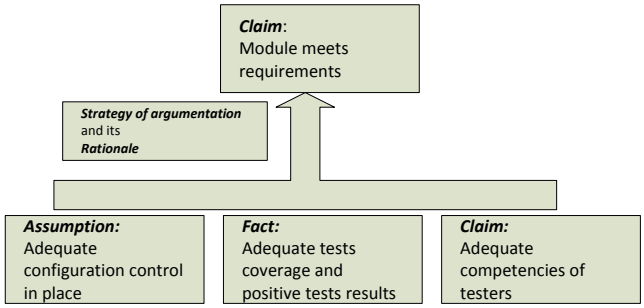

Figure 3 Extended *Module correctness argument*

Convincing arguments can be used to build trust, because they demonstrate trustworthiness. Such arguments we call *trust cases.* For example, a convincing (supported by evidence) argument that *a service is secure* increases trust in the service. The evidence supporting such argument could include: protective security measures used, certification procedures passed, penetration tests results, operating data, development practices used and so on.

In such case, the strategy of argumentation is modified to: *argumentation by referring to test results, test coverage and testers' competencies with the assumption that adequate configuration control is in place.*

In our research we have particular interest in two different types of trust cases: *assurance cases* which focus on demonstrating assurance of some chosen (and considered important) property (like safety, security, privacy, dependability, reliability etc.) and *conformance cases* where the focus is on demonstrating conformity with some predefined set of requirements (given in standards, norms, directives, regulations etc.).

The primary objects of interest for developing trust cases are ICT products, services and processes, however the scope of applicability of trust cases it very broad and includes all situations where human or technical objects establish trust relationships by exchanging arguments demonstrating their mutual trustworthiness.

3 Argument representation

The main challenge is to decide about the argument model and the corresponding language of expressing arguments to provide for adequate expressive power, understandability and scalability of arguments.

The model used in NOR-STA is presented in Figure 4.

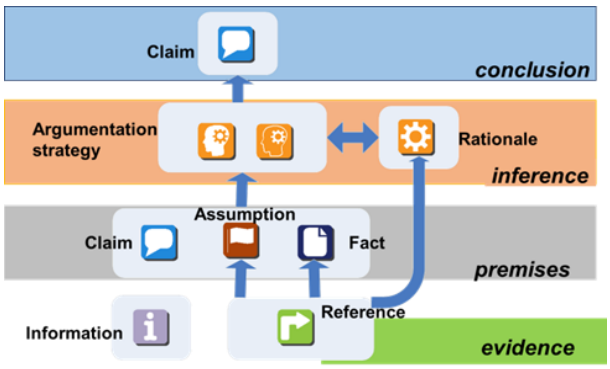

Figure 4 Argument model in NOR-STA

According to the model of Figure 4, the nodes (elements) of an argument are represented by different graphical icons. The icons can have textual descriptions (fitting to a single line) and in addition can have richer descriptions accessible after selecting a given node. The hierarchy of argumentation develops from left to right, as a set of structured lines, each line marked by a proper icon. For instance, the example *Module correctness argument* can be, following the model of Figure 4, represented as in Figure 5.

4 Communication and co-operation

To fulfil their role of supporting building and establishing trust, arguments need to be easily communicated between the interested parties. This leads to the requirement of controlled argument sharing with the objective to provide

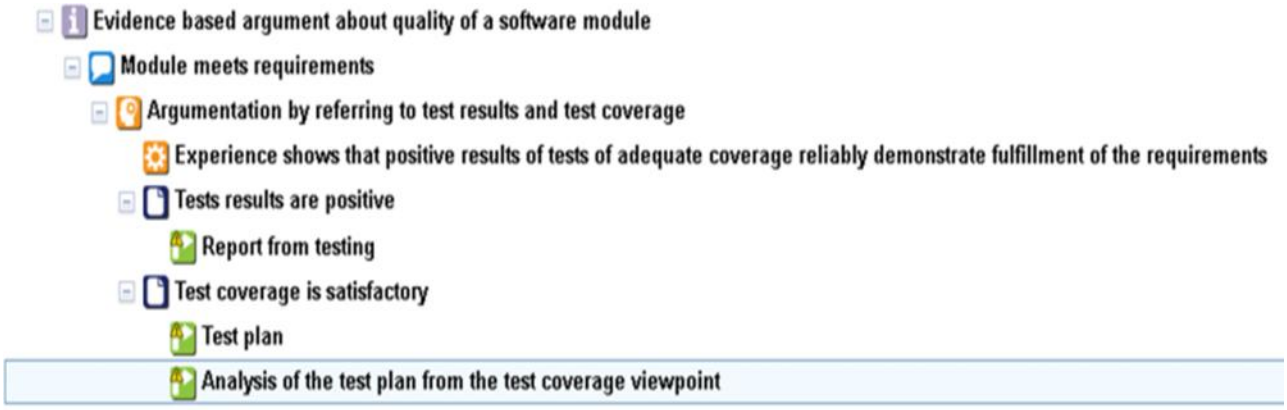

Figure 5 ***Module correctness argument*** **in NOR-STA**

easy access by the authorized parties and simultaneously to provide adequate protection against unauthorized accesses. Different roles can be identified while accessing an argument, for instance argument developer, argument assessor, argument viewer, argument administrator and so on. Each of these roles can be refined according to the needs, for instance we can distinguish different sub-roles of argument developer: those responsible for logic part of the argument and those responsible for the epistemic part (suppliers of evidence). Different roles may also be associated with different views at the argument, for instance an auditor of a conformance case can see the standard requirements and the associated evidence in a form which best supports his/her task of assessing conformance with the standard.

The above considerations led to key decisions related to the NOR-STA system:

- Deploying NOR-STA in accordance with the SaaS (Software-as-a-Service) model.
- Managing access control in accordance with the RBAC (Role-Based Access Control) model.
- Providing different views to support different roles the users play with respect to a given argument.

5 Argument assessment

Argument assessment is necessary in different scenarios, like decision making, consensus building or disputes resolution.

Both, logic and epistemic parts of an argument are subjected to assessment. The assessment involves appraisal of the 'compelling power' of an argument. The assessment results can be selected from a two-value scale {*accept*, *reject*} like in case of a mathematical proof, or from a more complex space distinguishing different levels of acceptance/rejection and the related uncertainty. Consequently, we can have different *argument assessment mechanisms* which can be applied with respect to the same argumentation structure.

Referring to our *Module correctness argument*, the logic related question is: *do successful tests of right coverage really demonstrate that the module meets its requirements?*. And the epistemic question is: *do we really have positive test results and do the tests adequately cover the requirements?*

Answering positively to the logic question we confirm that meeting the requirements by a software module can be demonstrated by developing an adequate test plan, running the corresponding tests and receiving positive results of the tests. Note that if accepted, such argumentation strategy can be reused with respect to other software modules as well.

If we still doubt about the answer to the logic question, we can modify the argumentation strategy by adding additional premises. For instance, in case of our example argument, the additional premises could be: ***Assumption:*** *adequate configuration management in place* and ***Claim:*** *adequate competencies of testers* (as shown in Figure 3).

Answering positively to the epistemic question means that satisfactory evidence has been provided demonstrating that the assertion (represented by a given fact) is true in the considered context. For instance, the fact: *test results are positive* can be demonstrated by providing the report from tests whereas the assertion *test coverage is adequate* can be demonstrated by providing the test plan and the result of the analysis of this plan against the relevant set of requirements.

Depending on the applied assessment mechanism, the results of assessment are selected from different scales. An example of an advanced assessment mechanism can be the mechanism based on Dempster-Shafer belief functions implemented in NOR-STA, which supports the two dimensional space: ***Decision***={*rejectable*, *opposable*, *tolerable*, *acceptable*} and ***Confidence***={*sure*, *very high*, *high*, *low*, *very low*, *uncertain*} from which the assessor selects his/her assessments. The details of this assessment mechanism can be found in [3].

From the experience we have so far with the NOR-STA system, different application domains may require

different argumentation assessment mechanisms and therefore it is essential that the tools supporting application of evidence-based argumentation were able to switch between different mechanisms depending on a particular usage context. Presently, the NOR-STA system implements some nine different argumentation assessment mechanisms and its architecture is open to easily absorb the new ones, if needed.

Let us consider a task of assessing a complex argument (multiple levels of the argumentation hierarchy and a large number of facts supported by related evidence). In most cases assessment of the epistemic part can be split into a number of independent (local) assessments: each fact can be considered in isolation and the assessor assesses to which extent the submitted evidence supports this fact (for instance, to which extent the submitted report from testing demonstrates that the results of tests are positive). The assessment of the logic part can also be localized, i.e. each argumentation strategy can be assessed in isolation by looking at its conclusion and its premises. The problem becomes more complex if we try to propagate (local) assessments of facts towards the assessments of claims which depend on these facts. In case of more advanced assessment mechanisms, manual realisation of this task can be very laborious and error prone. The solution is to define the aggregation rules which can then be implemented and performed automatically. For instance, such rules for the Dempster-Shafer based mechanism implemented in NOR-STA are documented in [3]. Having the aggregation of local assessments automated, we can assess 'large' arguments with a reasonable effort (NOR-STA users have such experience with arguments up to several thousands of nodes).

Another important issue is the way of presenting the argument assessment results. As arguments are (mostly) exchanged between people, it is of particular importance that the assessment results are presented in a human-friendly way and that they support tasks performed by the users. In our experience, using colours to distinguish different values from the argument assessment scale proven to be particularly effective (basic colours: red meaning rejection, green meaning acceptance and yellow meaning uncertainty). These basic colours can be then mixed to distinguish more fine values, while using more advanced assessment mechanisms. This way of visualization not only communicates the overall assessment (the assessment of the top claim of the argument) but also provides for easy identification of the 'weak' parts of the argumentation and supports decisions concerning improvement of the considered argument.

6 Size and change management

Arguments can be complex structures composed of a large number of nodes and integrating large number of pieces of evidence. Argument can also have a long lifespan during which the argument is subjected to changes and modifications. For instance, consider a conformance argument demonstrating that a given organisation is conformant with ISO27001 or a safety argument related to an autonomous vehicle. The scope of changes will include both, the structure and the evidence and the arguments will be subjected to different assessments (for instance, self-assessment, third party assessment, repeated assessment after certificate expiration and so on).

6.1 Operating 'large' arguments

Large arguments are difficult to handle and to understand (what does it mean 'large'? From NOR-STA users we have reports about arguments up to 8000 nodes). Representing hierarchies of this size in a graphical form causes problems with visualizing the hierarchy within the limits of a computer screen, inserting textual descriptions in graphical symbols and showing dependencies between nodes in a readable way. The NOR-STA way of representing the hierarchy from left-to-right (instead of from top-to-bottom) and representing each node in a single line is advantageous for large arguments (an analogy can be the commonly accepted way of presenting file directory structure as the left-to-right hierarchy instead of presenting it as a vertical graphical structure).

6.2 Managing massive evidence

A realistic argument (for instance, demonstrating conformance with a selected standard or demonstrating safety of a new technology to be applied off-shore) will integrate many different documents which contain evidence supporting the argumentation. These can be electronic documents of any format (textual, graphics, video, audio etc.) and the documents can reside in different locations with different access protocols (web pages, ftp, svn and so on). It is necessary to access these documents either in their target repositories or, alternatively, to provide for dedicated and adequately protected customized repositories. In many cases, the documents can be large (for instance, a design documentation of a medical device) and the evidence we want to refer to is a selected part of such document. In such case it is advantageous to have a possibility to refer to this particular part instead of referencing the whole document.

Often the evidence referred to in the argumentation is subjected to stringent security constraints (examples are personal data, trade secrets, reputation related data and so on). Therefore, while providing support for evidence-based arguments it is necessary to implement and to demonstrate conformance with (sometimes very demanding) security objectives which need to be met and continuously maintained.

6.3 Change management

Argument structure, the supporting evidence and argument assessments can be subjected to changes and modifications. This results in a continuous evolution of the whole argumentation and calls for an adequate change management mechanisms.

At the NOR-STA tool level, the following mechanisms proven their usefulness.

Accountability of changes where each modification introduced to the argumentation structure and to the assessments is recorded in the argument history providing for the identification of the responsible user.

Baseline mechanism where baseline is a (named) 'snapshot' of the current state of the whole argument. Such baseline can be later used as a well-defined reference (for instance, a full contents of the conformance argument which has been third-party assessed to obtain a formal certificate).

Rollback mechanism which provides access to the full history of changes and enables to roll-back to any previous moment from the history, if necessary (for instance, to choose an alternative way of developing the argument or to recover from a disaster).

7 Fitting into user business context

Assurance and conformance arguments have multiple stakeholders and it is important that these stakeholders can access the argument with their corresponding access rights. Therefore managing different user accounts and user access rights is necessary and the role of administrator of these accounts needs to be distinguished.

The users may maintain multiple arguments where each argument can have a different concern (for instance, conformance cases related to different standards, assurance arguments related to different products, arguments related to different objectives and so on). Therefore, it is necessary to provide for different 'working spaces' of different arguments and to support grouping arguments according to different criteria (for instance, different products, different standards, different assurance objectives, different organizational departments and so on).

The NOR-STA system supports users and user rights management and provides means for introducing structure into the set of different arguments. Each argument is maintained in its *project* (a sort of 'working space') where it undergoes its evolution. The projects can be arbitrarily organized into *folders* and the folders structure is hierarchical (resembling the file directory structure of operating systems). This mechanism provides for sufficient flexibility of organizing different arguments into a structure which meets expectations of user organisation. Purposeful grouping of projects into folders also helps in enforcing common access policies with respect to 'similar' arguments.

Assurance (or conformance) case can be treated as an electronic document which maintains a convincing argumentation that the user organization achieves some important objectives (for instance, meeting the safety requirements by its product or being conformant with selected standards). In the present business contexts however, it is often required that such information is presented in a more 'traditional' form, for instance as printed documents of predefined structure. Therefore the issue of reports generation cannot be neglected.

The solution applied in NOR-STA is to provide for a number of pre-defined reports of the metrics related to an argument and reports presenting the contents of the whole argument (in pdf and in doc formats) and in addition to this to provide for integrating with commonly accessible tools, like Excel, which support different forms of data presentation. Integration through XML/HTML data and XLS scripts to process XML data provide for high flexibility in generating reports in different structures and formats.

Figure 6 presents an example report generated from NOR-STA, where the *Module correctness argument* (shown in Figure 5) is presented in the GSN notation [4].

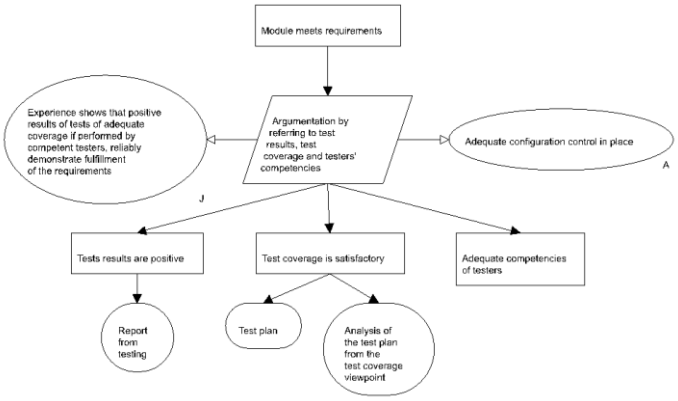

Figure 6 GSN representation of *Module correctness argument*

8 Integration

Evidence-based argument is not being used in isolation. Instead, it has to be integrated within the broader context to which the argument is expected to bring an added value. In particular, this context can include other systems supporting the users' tasks and various repositories which store documents that are vital to achieving business objectives of the user organisation. These documents, produced by business processes (Design, V&V, QA, HR and others) are the sources of evidence which is referred to by the argument demonstrating achievement of the assurance/conformance goals of the organisation.

In NOR-STA system, the key to integration with other systems is the API (Application Program Interface) implemented as a set of web services which cover full functionality of NOR-STA. This provides a technical base for integration with selected external systems or services, according with the needs. Examples are Single Sign On (SSO), Active Directory Federation Services (ADFS), Azure B2C or Siemens Teamcenter.

Another technical base for integration is XML based export/import of the whole argument which can then be processed by dedicated applications, if needed. In particular, this provides for argument contents exchangeability with other tools supporting evidence-based argumentation.

9 Argument structuring and reuse

Evidence-based arguments can be structured following different (not necessarily orthogonal) decomposition

criteria. Examples are: risks based decomposition where the argument addresses relevant risks and demonstrates that they are adequately mitigated or architecture/design model based decomposition where the argument follows the structure of the considered system, its subsystems and modules and demonstrates their selected properties.

More support is needed for automatic derivation of assurance case structure from the results of (standardized) risk analyses [5] or from the architectural/design models of a system [6].

Argumentation reuse has the potential of significant reduction of development effort by standardization of typical substructures recurring in arguments. A NOR-STA represented inventory of *design patterns* of arguments can be found in [7]. A particular pattern supporting the reuse of conformance cases is called *conformance template* [8]. This is the logic part of the argument which reflects the structure of the requirements of a selected standard. As long as the corresponding standard remain unchanged this logic part can be reused in in each conformance case which demonstrates conformity with the standard.

If the standard changes, however (and all 'living' standards undergo evolution), the changes need to be reflected in the conformance template and then propagated to all conformance arguments that were created following the template. NOR-STA supports such automatic propagation of changes introduced to a conformance template. The intention is to maintain consistency between the (changing) standard and its (multiple) applications in various target contexts.

10 Composability

Assurance/conformance cases are being used in different contexts. For instance, a component produced by its manufacturer is being delivered with its assurance case and after being sold to different buyers, is used in different systems. The developer of the assurance case of the target system would be interested to refer to the component assurance case and to reuse its assessment results. The questions arising in such scenario include: how to interface the component related assurance case to the system related assurance case, how to pass the assessment results of the component related assurance case and what if this result is context dependent (the assessment of the component related assurance case can be different depending on the target system context) and so on.

Presently, in NOR-STA system there are two mechanisms supporting composability of assurance/conformance cases: explicit representation of assumptions and *required/provided interfaces*.

Distinguishing a separate node type for representing assumptions (see Figure 4) provides for explicit enumeration of the assumptions conditioning a given evidence-based argument and protects against assumptions overlooking and omissions. In Figure 3 we have an explicit assumption that the module is being tested assuming that *adequate configuration control is in place* which prevents against situations where, for instance, the tests were performed according to an invalid test plan. While using the *Module correctness argument* within the context of the system embedding the module, we can verify if this assumption is still valid before accepting the result of the assessment of this argument.

NOR-STA also supports explicit declarations of interfaces between assurance/conformance case components. Consider an extended version of the assurance case of our example software module shown in Figure 3. The premise *adequate competencies of testers* is a claim which needs to be further demonstrated. Assume that this claim has been demonstrated by a separate *Tester competencies argument* which by declaring its *provided interface* make this claim and its assessment visible to the outside world. Inside the module, the claim is demonstrated by, for instance, using CV-s of the testers as the supporting evidence.

If the argument shown in Figure 3 declares as its *required interface* the claim *adequate competencies of testers* and if the two interfaces (the provided one and the required one) are *bound* together, then the two modules (the *Module correctness argument* and the *Tester competencies argument*) form a single argument independently of if the *Tester competencies argument* is also used in other contexts. The results of the assessment of the tester competencies will be automatically propagated to each assurance case which is bound with the *Tester competencies argument* through the provided/required interfaces.

11 Conclusion

Argument is a focal point situated between different stakeholders and addressing their concerns. By exchanging arguments the users can develop mutual trust that their concerns are being addressed with the satisfactory assurance.

In this presentation we have briefly characterized some of the main challenges and the related decisions which were made during development and deployment of NOR-STA – a system supporting development, assessment and maintenance of evidence-based arguments in different application contexts.

Presently, NOR-STA is used commercially in different domains, including medical, oil and gas, automotive, flight control and others. The short-term strategy of further development is customer-driven and follows the needs of current and future users. Equally important is also the long-term strategy which looks into the trends and tries to identify the future challenges, even if not yet articulated by the present customers. Two challenges can be considered as examples.

Firstly, better support for composability of arguments, not only at the syntactic level (provided/required interfaces) but also at the semantic level (matching the contexts within which arguments maintain their validity).

Secondly, continuous assessment of an argument which follows the changes in the evidence and automatically

reflects these changes in the assessment of the argument. This could for instance support the concept of *continuous certification* as opposed to the present practices of repeated certification based on a predefined schedule (which is being criticized as inadequate for, for instance, security certificates in a very dynamically changing landscape of security threats).

Acknowledgement

Research, development and commercial deployment of NOR-STA had multiple contributors. In particular, the contribution of the following colleagues is to be acknowledged (in alphabetic order): dr Łukasz Cyra, Jakub Czyżnikiewicz (programmer), dr Aleksander Jarzębowicz, dr Jakub Miler, dr Andrzej Wardziński, Michał Witkowicz (programmer).

References

[1] Arevide sp. z o.o., www.argevide.com

[2] *Challenges in providing support for management of evidence based arguments*, https://www.argevide.com/wp-content/uploads/2016/05/Argevide-WP3-Challenges.pdf

[3] L. Cyra, and J. Górski (2011), *Support for Argument Structures Review and Assessment*, Reliability Engineering and System Safety (96), Elsevier, pp. 26-37.

[4] Goal Structuring Notation Community Standard, Version 2 (2018), https://scsc.uk/r141B:1?t=1

[5] A Wardzinski and P Jones (2017), *Uniform Model Interface for Assurance Case Integration with System Models*, Computer Safety, Reliability, and Security, Springer, pp. 39-51.

[6] R Hawkins, I Habli, D Kolovos, R Paige, T Kelly (2015), *Weaving an Assurance Case from Design: A Model-Based Approach*, IEEE Xplore.

[7] M Szczygielska and A Jarzębowicz (2018), *Assurance Case Patterns On-line Catalogue*, In: Advances in Dependability Engineering of Complex Systems. Springer, IND 141625.